

Nom Prénom	
------------	--



- Classes -

TP numéro 6

Objectif du TP	Prise en main de l'utilisation des classes
Pré requis	Cours
Ressources à disposition	EDI
Capacités / compétences visées	Maîtriser l'utilisation des classes
Travail demandé	Lire l'énoncé, appliquer les consignes
Suivi du professeur	Suivi du déroulement du travail des étudiants Passage auprès de chaque étudiant plusieurs fois dans le TP. Aide personnalisée. Réponse aux questions
Durée du TP	1 heure
Bilan / observations	

Contenu

1. Classe Domino	2
2. classe CompteBancaire	2
3. Classe Voiture	3
4. Classe CompteEpargne	4

1. Classe Domino

Définissez une classe **Domino()** qui permette d'instancier des objets simulant les pièces d'un jeu de dominos. Le constructeur de cette classe initialisera les valeurs des points présents sur les deux faces A et B du domino (valeurs par défaut = 0).

Deux autres méthodes seront définies :

- une méthode **affiche_points()** qui affiche les points présents sur les deux faces ;
- une méthode **valeur()** qui renvoie la somme des points présents sur les 2 faces.

Exemples d'utilisation de cette classe :

```
>>> d1 = Domino(2,6)
>>> d2 = Domino(4,3)
>>> d1.affiche_points()
face A : 2 face B : 6
>>> d2.affiche_points()
face A : 4 face B : 3
>>> print("total des points :", d1.valeur() + d2.valeur())
15
>>> liste_dominos = []
>>> for i in range(7):
...     liste_dominos.append(Domino(6, i))
>>> print(liste_dominos[3])
<__main__.Domino object at 0xb758b92c>
```

etc.

2. classe CompteBancaire

Définissez une classe **CompteBancaire()**, qui permette d'instancier des objets tels que **compte1**, **compte2**, etc. Le constructeur de cette classe initialisera deux attributs d'instance **nom** et **solde**, avec les valeurs par défaut 'Dupont' et 1000.

Trois autres méthodes seront définies :

- **depot(somme)** permettra d'ajouter une certaine somme au solde ;
- **retrait(somme)** permettra de retirer une certaine somme du solde ;
- **affiche()** permettra d'afficher le nom du titulaire et le solde de son compte.

Exemples d'utilisation de cette classe :

```
>>> compte1 = CompteBancaire('Duchmol', 800)
>>> compte1.depot(350)
>>> compte1.retrait(200)
>>> compte1.affiche()
Le solde du compte bancaire de Duchmol est de 950 euros.
>>> compte2 = CompteBancaire()
>>> compte2.depot(25)
>>> compte2.affiche()
Le solde du compte bancaire de Dupont est de 1025 euros.
```

3. Classe Voiture

Définissez une classe **Voiture()** qui permette d'instancier des objets reproduisant le comportement de voitures automobiles. Le constructeur de cette classe initialisera les attributs d'instance suivants, avec les valeurs par défaut indiquées :

marque = 'Ford', couleur = 'rouge', pilote = 'personne', vitesse = 0.

Lorsque l'oninstanciera un nouvel objet **Voiture()**, on pourra choisir sa marque et sa couleur, mais pas sa vitesse, ni le nom de son conducteur.

Les méthodes suivantes seront définies :

- **choix_conducteur(nom)** permettra de désigner (ou changer) le nom du conducteur.
- **accelerer(taux, duree)** permettra de faire varier la vitesse de la voiture. La variation de vitesse obtenue sera égale au produit : **taux × duree**. Par exemple, si la voiture accélère au taux de 1,3 m/s pendant 20 secondes, son gain de vitesse doit être égal à 26 m/s. Des taux négatifs seront acceptés (ce qui permettra de décélérer). La variation de vitesse ne sera pas autorisée si le conducteur est 'personne'.
- **affiche_tout()** permettra de faire apparaître les propriétés présentes de la voiture, c'est-à-dire sa marque, sa couleur, le nom de son conducteur, sa vitesse.

Exemples d'utilisation de cette classe :

```
>>> a1 = Voiture('Peugeot', 'bleue')
>>> a2 = Voiture(couleur = 'verte')
>>> a3 = Voiture('Mercedes')
>>> a1.choix_conducteur('Roméo')
>>> a2.choix_conducteur('Juliette')
>>> a2.accelerer(1.8, 12)
>>> a3.accelerer(1.9, 11)
Cette voiture n'a pas de conducteur !
>>> a2.affiche_tout()
Ford verte pilotée par Juliette, vitesse = 21.6 m/s.
>>> a3.affiche_tout()
Mercedes rouge pilotée par personne, vitesse = 0 m/s.
```

4. Classe *CompteEpargne*

Écrivez un nouveau script qui récupère le code de l'exercice 2 (compte bancaire) en l'important comme un module. Définissez-y une nouvelle classe **CompteEpargne()**, dérivant de la classe **CompteBancaire()** importée, qui permette de créer des comptes d'épargne rapportant un certain intérêt au cours du temps. Pour simplifier, nous admettrons que ces intérêts sont calculés tous les mois. Le constructeur de votre nouvelle classe devra initialiser un taux d'intérêt mensuel par défaut égal à **0,3 %**. Une méthode **changeTaux(valeur)** devra permettre de modifier ce taux à volonté. Une méthode **capitalisation(nombreMois)** devra :

- afficher le nombre de mois et le taux d'intérêt pris en compte ;
- calculer le solde atteint en capitalisant les intérêts composés, pour le taux et le nombre de mois qui auront été choisis.

Exemple d'utilisation de la nouvelle classe :

```
>>> c1 = CompteEpargne('Duvivier', 600)
>>> c1.depot(350)
>>> c1.affiche()
Le solde du compte bancaire de Duvivier est de 950 euros.
>>> c1.capitalisation(12)
Capitalisation sur 12 mois au taux mensuel de 0.3 %.
>>> c1.affiche()
Le solde du compte bancaire de Duvivier est de 984.769981274 euros.
>>> c1.changeTaux(.5)
>>> c1.capitalisation(12)
Capitalisation sur 12 mois au taux mensuel de 0.5 %.
>>> c1.affiche()
Le solde du compte bancaire de Duvivier est de 1045.50843891 euros.
```